

# Requirement Document



# Characteristics of good system analyst

- **Impertinence**: you should ask as much questions as possible to collect the required information. You should ask the questions involving any unclear parts of the system during the interview.
- **Impartiality**: You must find the best organizational solution to a business problem or opportunity. You should not try to impose yourself, you must examine all the ideas and then make a decision which solution you will use.
- **Relaxation of constraints**: You must assume that everything is possible and eliminate the infeasible (i.e. not applicable).

# Characteristics of good system analyst

- **Attention to details:** You must make notes of every detail said at interview or was written in a form.
- **Reframing:** You must not jump to conclusions about the built system. You must view the system as the users requirements tell you not as you have seen in a previous system unless the users want that.

# Requirements Specification

- Make the requirement document that will be used in design and as a guide to show to the users.

## Types of requirement document

- User requirements
  - Statements written in natural language (NL) plus diagrams of the services the system provides and its operational constraints. Written for customers.
- System requirements
  - A structured document setting out detailed descriptions of the system services. Written as a contract between client and contractor.
- Software specification
  - A detailed software description which is used for design or implementation. Written for developers

# How to write the Requirement Document?

## Part 1: Introduction

- It describe the SW to be built in brief giving a brief description of it and its functions (services), the name of company that it is built for, the goal of building such system. 1-3 pages.

## Part 2: Hardware

- Describes the HW that this SW will run on (TV, Washer, Mobile).
- If using on-self HW, it describes the minimum configuration that this SW will work under (RAM size. Processor speed, Hard Disk free size, monitor resolution, network connection,.....)

# How to write the Requirement Document?

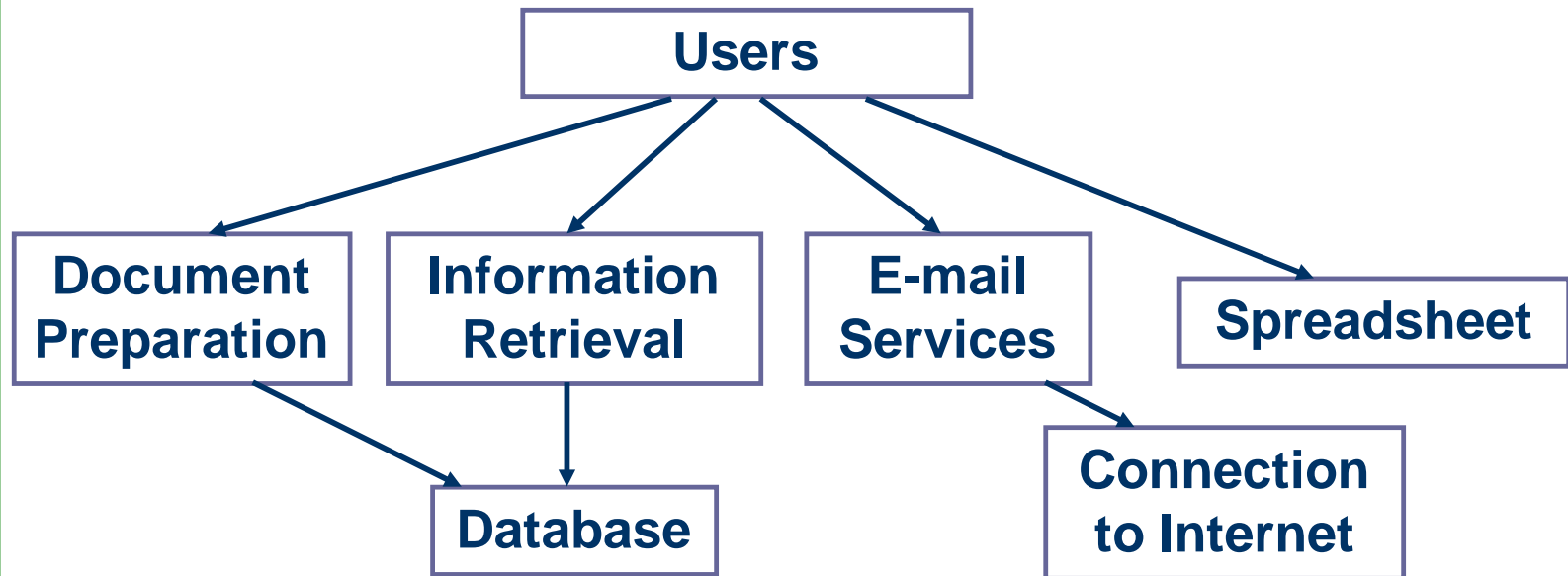
## Part 3: Conceptual Model

- It is a graphical representation of the SW services required, their relationship, and any external HW or SW requirements connected to these services.

## Example:

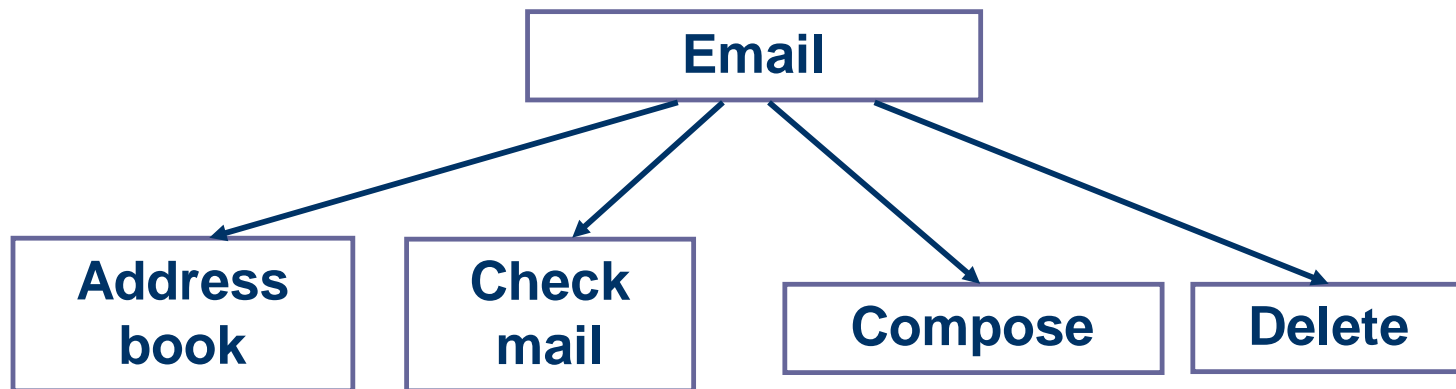
- If a company need a SW that performs the following functions; Document preparation (word processor), Information retrieval (in Database), E-mailing, Spread sheet (Excel)

## Conceptual Model for the example



Each of these services can then be described in more details in a separate figure

## Conceptual Model for one of the services (e-mail) in the example



**The number of the figures of the conceptual model of the SW Implies the size of SW and also its complexity**



## Part 4: Functional requirements

- It fully describes the functions (services) required by the SW and drawn in the conceptual model.
- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- Functional user requirements may be high-level statements of what the system should do
- But functional system requirements should describe the system services in detail

# Functional Requirement for part of the example

- 1. E-mail function

- **1.1 Address book:** it is used to store the important addresses. Then when composing mail, the user can use it to insert address.
- **1.2 Check mail:** it is used to access the mail storage area and to find the unread messages.
- .....

## Other Examples of functional requirements

### Ex 1:

- The user must be able to search either all of the of databases or select a part from it to search in.

### Ex 2:

- The system must provide the user with the ability to read documents stored.

### Ex3:

- Every order must be allocated a unique identifier (ORDER\_ID) which the user will be able to copy to the account's permanent storage area.

# How to write the functional requirement part?

## 1- Using natural language (NL)

### Problems with natural language:

- Presence of excess noise: the presence of parts of text that could be viewed as noise as it does not contain information relative to the function.
- Precision is difficult without making the document difficult to read.
- Requirements confusion

# Problems with natural language

- **Functional and non-functional requirements tend to be mixed-up**
- **Several different requirements may be expressed together**
- **It can neglect important details that is necessary to understand the function**

## 2- Using Formal or semi-formal language

- **Formal**: as in programming language. Not available till now.
- **Semi-formal (structured)**: Impose some structure on the way we should write the functional requirement. Like templates or forms that should be filled.
- Thus, it limits the NL parts written.
- Some of these languages use graphs to express the functions and their interrelationship.

# Using semi-formal languages

## Example of a template:

- Name of the function:-----
- Task to be made: -----
- Produced from previous function:-----
- Has children functions:-----
- Inputs to function:-----
- Output from function:-----
- -----

# Examples of semi-formal languages

- **1) PSL/PSA:** Programmable structured language/programmable structured analysis
- **2) RSL:** Requirement specification language. It also has a compiler that accepts the structured form and generates a NL version to show to the user.
- **3) SADT:** Structured analysis developing tool. Uses structured form and graphical symbols.



## **Part 5: Non-functional requirements**

- **Define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.**
- **Process requirements may also specify a particular programming language or development method**
- **Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless**

## **Part 6 - Database requirements**

- This part will contain the database requirements if the system uses a database.
- We should include the entities required to be stored and the relations between them represented as E-R model.
- The type of each data field must also be determined (i.e. integer, text,...), the length of text fields, and any restrictions imposed on any field (i.e. non zero value, automatically generated number,...).